**Real Application Clusters**

云和恩墨(北京)信息技术有限公司

技术顾问 燕鑫

http://www.enmotech.com

**文档控制：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序** | **版本号** | **更改人** | **日期** | **备注** |
| 1 | 1.0版 | 燕鑫 | 2018-11-1 | 初始版本 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. RAC概要 - 5 -](#_Toc530667412)

[1.1 RAC概要 - 5 -](#_Toc530667413)

[1.2 RAC One Node概要 - 6 -](#_Toc530667414)

[1.3 Oracle Clusterware概要 - 7 -](#_Toc530667415)

[1.3.1 Flex Clusters - 7 -](#_Toc530667416)

[1.3.2 Reader Nodes - 7 -](#_Toc530667417)

[1.3.3 Local Temporary Tablespaces - 8 -](#_Toc530667418)

[1.4 RAC架构和进程概要 - 8 -](#_Toc530667419)

[1.4.1 Cluster-Aware Storage解决方案 - 8 -](#_Toc530667420)

[1.4.2 Network Connectivity - 8 -](#_Toc530667421)

[1.4.3 Dynamic Database Services - 9 -](#_Toc530667422)

[1.4.4 Virtual IP Addresses - 9 -](#_Toc530667423)

[1.4.5 Restricted Service Registration - 9 -](#_Toc530667424)

[1.4.6 Software Components - 10 -](#_Toc530667425)

[1.4.7 Background Processes - 11 -](#_Toc530667426)

[1.5 使用服务实现自动负载均衡 - 12 -](#_Toc530667427)

[1.6 Server Pools and Policy-Managed Databases - 14 -](#_Toc530667428)

[1.7 Quality of Service Management - 14 -](#_Toc530667429)

[1.8 Hang Manager - 14 -](#_Toc530667430)

[1.9 Multitenant with Oracle RAC - 15 -](#_Toc530667431)

[1.10 Database In-Memory and Oracle RAC - 15 -](#_Toc530667432)

[1.11 Managing Oracle RAC Environments - 15 -](#_Toc530667433)

[1.11.1 Administrative Tools - 15 -](#_Toc530667434)

[1.11.2 Monitoring - 16 -](#_Toc530667435)

[1.11.3 Evaluating Performance - 16 -](#_Toc530667436)

[2. Administering Storage in Oracle RAC - 16 -](#_Toc530667437)

[3. 管理数据库实例和Cluster Databases - 16 -](#_Toc530667438)

# RAC概要

## RAC概要

非集群的数据库架构中数据库与是实例的关系是一对一，而在集群中，数据库与实例的关系是一对多。一个RAC集群中，最多可以有100个实例。

与单实例数据库不同，RAC中每一个实例都有自己的：

1. redo thread
2. undo表空间

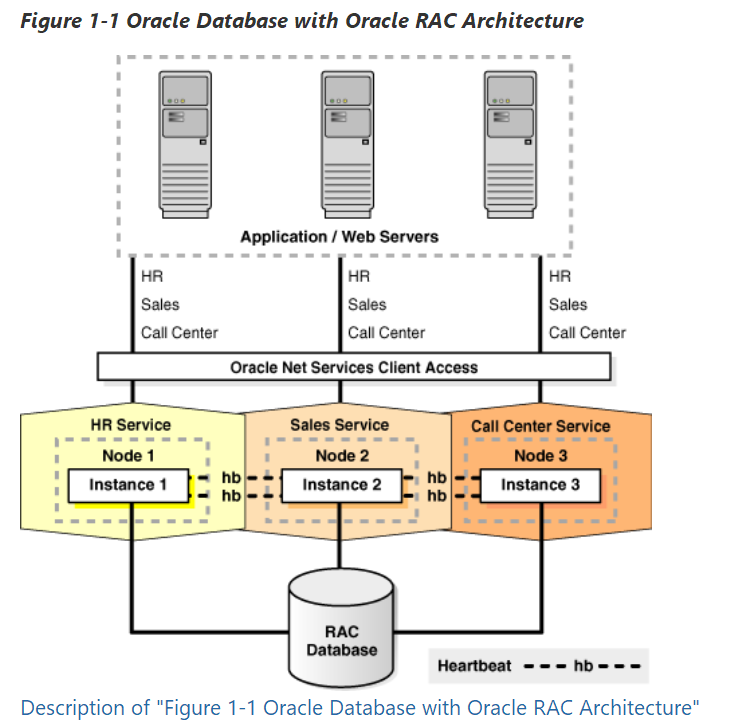
一个集群就是多台内部相连的服务器组成的一个集合，但是对于终端用户及应用来说，就如同在使用一台服务器。

Oracle Clusterware就是与ORACLE数据库整合在一起的集群管理软件，可以为RAC的使用提供基础的架构。

Oracle Clusterware会管理例如VIP（Virtual Internet Protocol）地址、数据库、监听、服务等等。

当然，Oracle Clusterware也可以为非集群（单实例）数据库提供高可用架构。因为，Oracle Clusterware+ASM构成的Oracle Grid Infrastructure可以同时为单实例或者RAC提供服务。

下图展现了ORACLE RAC是一样将多台服务器以一个single system image的方式对外呈现的，注意每一个instnace都必须拥有自己单独的服务器：



我们可以看到，集群较单实例的优势就是：

1. 多个服务器的联合处理能力可提供更大生产量
2. 更好的伸缩性，某一个节点实例失败，并不会对数据库的使用造成致命影响。

通常来说ORACLE RAC是部署在一个data center中的，也就是说，所有RAC中服务器的位置地理上是在一起的（同一个机房）。但是有时候出于安全需要，希望将一些节点在地理上与另外一些节点分开放置，比如不在一个机房、不在一栋楼、甚至不在一个国家。这是就可以使用Oracle Extended Cluster。但是因为数据必须同时存在与不同地点，那么就需要使用存储的镜像技术。

虽然Extended Cluster比传统RAC容灾性更好，但是节点间的通信延迟将成为问题，所以，官方更推荐使用DG来达到异地容灾的需求，而不是使用Extended Cluster。

## RAC One Node概要

## Oracle Clusterware概要

Oracle Clusterware可以满足所有管理一个集群数据库所需的要求：node membership，group services，global resource management以及high availability functions。

当我们搭建了一个RAC数据库时，数据库以及其VIP address，SCAN vip，SCAN listener， Oracle Notification Service和Oracle Net listeners会一同被注册到Oracle Clusterware中，并被Oracle Clusterware管理。

这些资源会随着节点上实例的启动而自动启动，并会在失败后自动重启。

Oracle Clusterware的守护进程会存在于每一个节点上。

所有被Oracle Clusterware管理的都叫做CRS resource。一个CRS resource可以是一个database， 一个instance，一个service，一个listener，一个VIP address或是一个application process。

Oracle Clusterware是通过存储在OCR（Oracle Cluster Registry）中的资源配置信息（resource's configuration information）来管理CRS资源的。

可以使用SRVCTL工具管理任何Oracle-defined的CRS资源。如果我们想管理那些非Oracle预定义的集群中的进程，我们可以使用Oracle Clusterware提供的框架（语法结构）创建相应的CRS资源来实现该目的。

### Flex Clusters

Flex Cluster主要应用于有大量节点的集群。该集群架构有俩种类型的节点：hub nodes和leaf nodes。Hub nodes最多可以是64个，leaf nodes可以更多。俩种类型的节点可以运行不同类型的应用。

### Reader Nodes

Reader nodes跟hub nodes有功能上的区别，hub nodes上的实例就跟之前版本的RAC集群的实例类似，而reader nodes上的实例就大不同了：

1. instance是read-only的
2. 重新配置Hub Node clusterware不会影响对reader nodes上的数据库实例的访问，只要hub nodes还能连接并且没被clusterware驱逐。一个hub node上可以有64个read node。
3. 因为是只读模式，所以不能使用DDL或者DML语句。

Oracle建议reader nodes上的memory尽量给大，这样开并行查询的时候效率更高。

### Local Temporary Tablespaces

## RAC架构和进程概要

Oracle RAC要求Oracle Clusterware能够提供

1. 使集群中的所有节点对相同存储的同时访问的功能架构
2. 使节点的进程之间能够进行内部通信的（IPC--Inter-Process Communication）的通信协议，使得所有节点上的实例可以同时处理数据，就好像数据在一个单个的cache上。
3. 监控集群中节点状态的机制。

### Cluster-Aware Storage解决方案

以为Oracle RAC数据库是属于shared-everything的数据库，所以必须将所有的datafile、controlfile、SPFILE、redolog放在集群识别的存储上（Cluster-Aware Storage），说白了就是每个节点都能访问得到的共享存储。

官方给出如下几种解决方案：

1. ASM
2. ACFS
3. 第三方集群文件系统，比如：Oracle OCFS2 (Linux, only)、IBM GPFS (IBM AIX, only)
4. NFS

### Network Connectivity

RAC环境中的所有节点都必须连接到至少一个局域网（Local Area Network -- LAN），从而使用户和应用可以访问RAC，这就是所谓的公网（public network）。

除了公网，RAC环境还需要一个私有网络，只供集群中各节点之间相互通信，就是所谓的私网（private network），或者说是interconnect。必须为私网配置至少一个交换机（switch）和千兆适配器（Gigabit Ethernet adapter）。

用户或应用切不可使用私网访问数据库实例，因为Cache Fusion要使用私网来进行实例间的内部通信。

RAC中实例间内部通信的私网可以使用UDP（User Datagram Protocol）协议或者RDS（Reliable Data Socket）协议。但是Oracle Clusterware的私网只能使用UDP协议，不能使用RDS协议。

如果使用NAS（Network Attached Storage），比如NFS，还需要额外的网络，该网络通道不能与公网或私网的网络通道公用。

### Dynamic Database Services

应用应当使用Dynamic Database Services来通过公网连接数据库。Dynamic Database Services可以让我们定义一个用户或一个应用连接数据库实例的一些规则和特征。

这些特征包括：unique name，workload balancing，failover options和high availability characteristics。

### Virtual IP Addresses

VIP就是节点的公网的虚拟IP，客户端可以通过VIP来访问各节点。

从一个客户端连接访问RAC的数据库实例的大体流程如下：

1. database client连接SCAN，提供给SCAN listener一个有效的服务名。
2. 然后SCAN listener来确定这个服务是被哪些instance拥有，然后决定把该请求路由给本地监听或者是另外的节点的监听。
3. 然后监听着某个符合条件的节点的VIP和对应端口的listner，接受到连接请求，最终将database client与该节点上的instance进行连接。

If a node fails, then the VIP address fails over to another node on which the VIP address can accept TCP connections, but it does not accept connections to the Oracle database. Clients that attempt to connect to a VIP address not residing on its home node receive a rapid connection refused error instead of waiting for TCP connect timeout messages. When the network on which the VIP is configured comes back online, Oracle Clusterware fails back the VIP to its home node, where connections are accepted. Generally, VIP addresses fail over when:

1. The node on which a VIP address runs fails
2. All interfaces for the VIP address fail
3. All interfaces for the VIP address are disconnected from the network

如果一个节点失败，然后VIP会failover到另一个节点，在该节点上VIP可以接受TCP连接，但它不接受与Oracle数据库的连接。尝试连接到已不在其自己的节点上的VIP地址的客户端会收到快速连接拒绝错误，而不是等待TCP连接超时消息。当发生漂移的VIP的自身节点的网络恢复正常后，Oracle Clusterware会将该VIP恢复回到其原本的节点上。通常，VIP地址在以下情况下进行漂移（failover）：

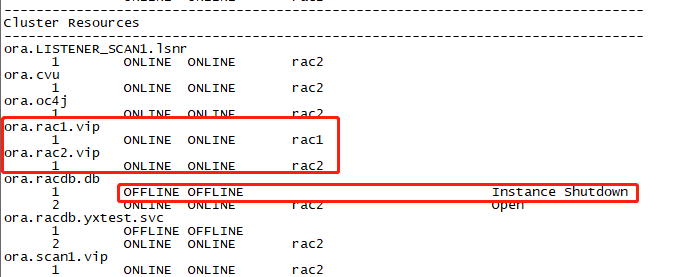
1. VIP自身的节点故障
2. VIP的所有网卡都故障
3. VIP的所有网卡都与网络断开连接
4. 正常关闭集群某个节点，该节点的VIP也会漂移到其他节点。

#### VIP故障转移实验

我们来看正常关闭的过程。

我们需要把俩个节点的系统日志（/var/log/messages）打开。

然后我们先正常关闭1节点数据库：



我们看到，vip还是在各自节点上。

然后我们正常关闭一节点的集群：

[root@rac1 ~]# /u01/app/11.2.0/grid\_11204/bin/crsctl stop crs

我先把我的/etc/hosts贴出来：

192.168.59.135 rac1

192.168.59.136 rac2

192.168.116.133 rac1-priv

192.168.116.134 rac2-priv

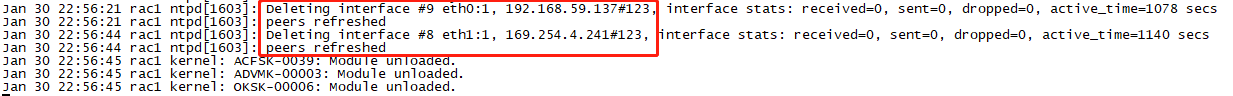
192.168.59.137 rac1-vip

192.168.59.138 rac2-vip

192.168.59.139 rac-scan

我们来看俩边的系统日志：

节点1如下：

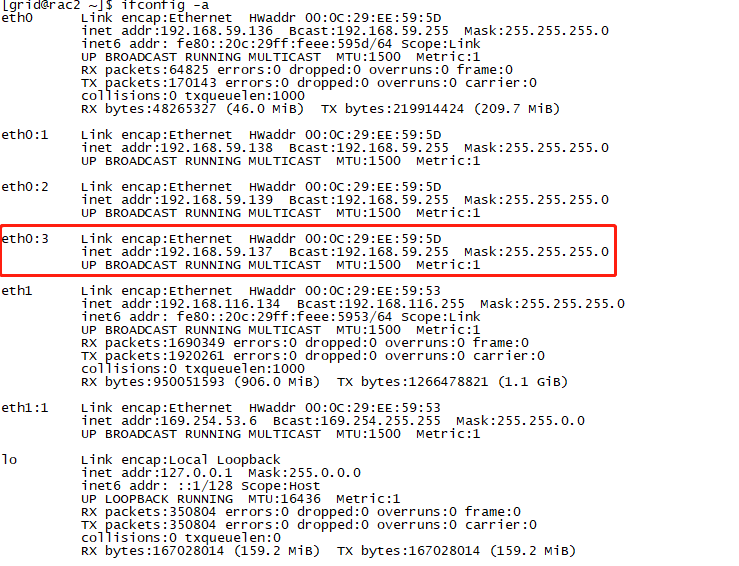


节点2如下：



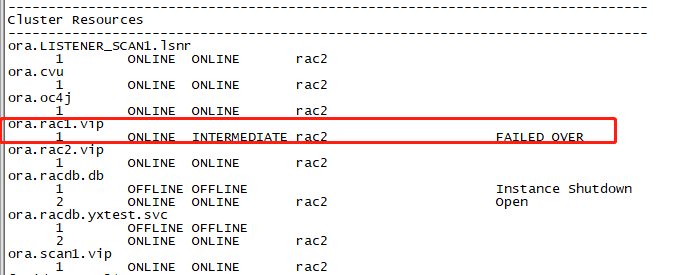
可以看到，节点1的VIP漂到了节点2的eth0上。

我们在节点二用ifconfig看一下：

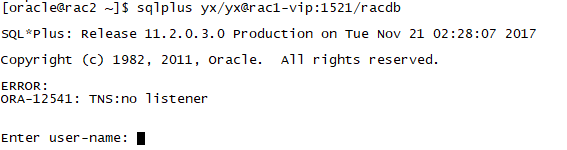


可以看到，确实节点1的VIP到节点2了。

在看集群资源：

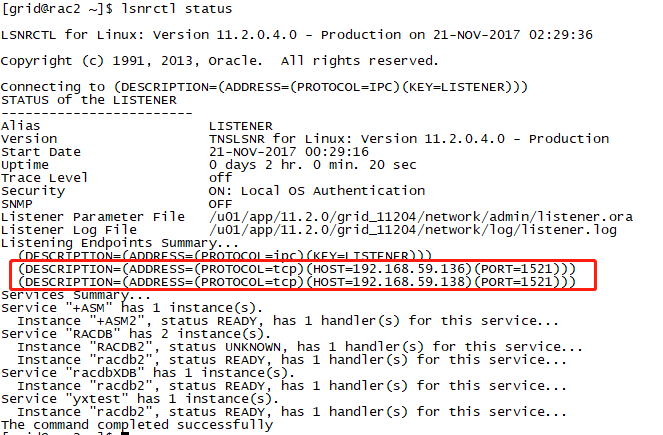


然后我们来尝试使用节点1的VIP连接数据库：



可以看到，报错是无监听。

我们看下节点2上的监听：



我们看到监听没有监听到这个地址。

那么我们该怎么用这个漂移过来的VIP连接数据库呢？我们需要修改监听配置。

我这里不用默认监听，用别的监听来实验：

LSNR2=

(DESCRIPTION\_LIST=

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.59.137)(PORT=1522)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.59.136)(PORT=1522)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.59.138)(PORT=1522)))

)

SID\_LIST\_LSNR2 =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = racdb)

(ORACLE\_HOME = /u01/app/oracle/product/11.2.0/dbhome\_1)

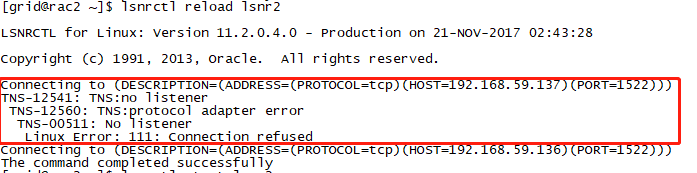
(SID\_NAME = racdb2)

)

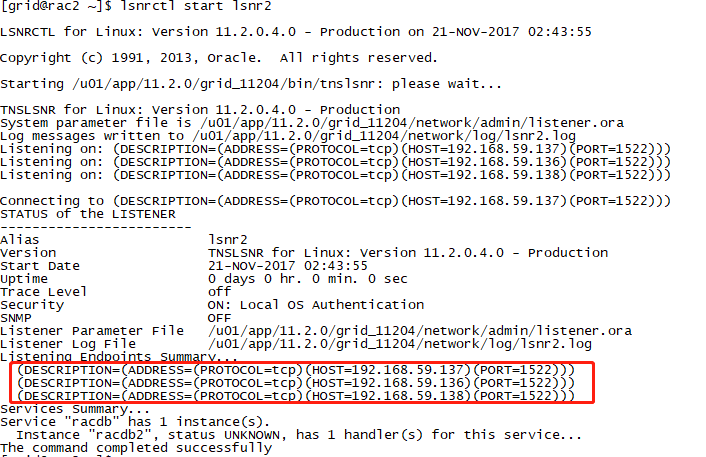
)

DIAG\_ADR\_ENABLED\_LSNR2=OFF

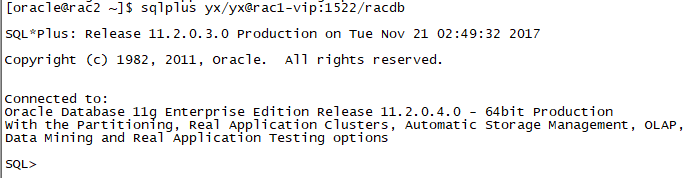
然后我们不能reload lsnr2，reload会报错，就是因为这个别人的VIP的原因，我们一定要重启监听。



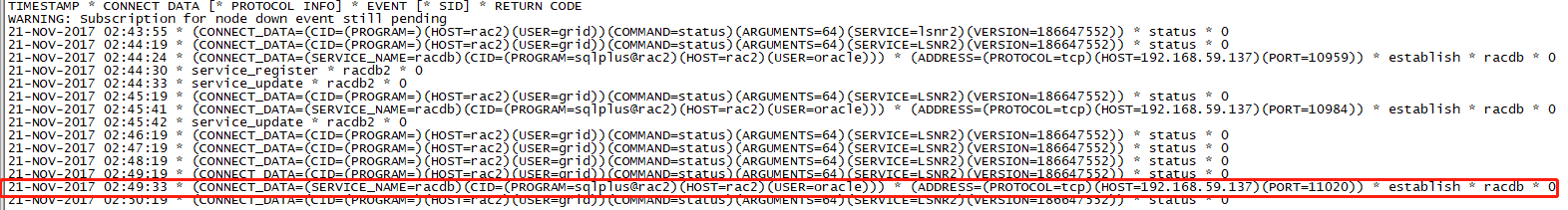
我们来start：



然后我们再用rac1-vip来连接数据库：



连接成功。监听日志也可以看到：



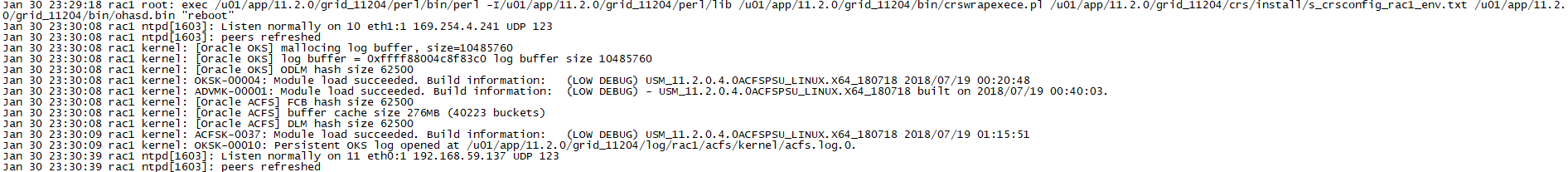
现在我们把一节点的crs重新启动，VIP会回到1节点上：

[root@rac1 ~]# /u01/app/11.2.0/grid\_11204/bin/crsctl start crs

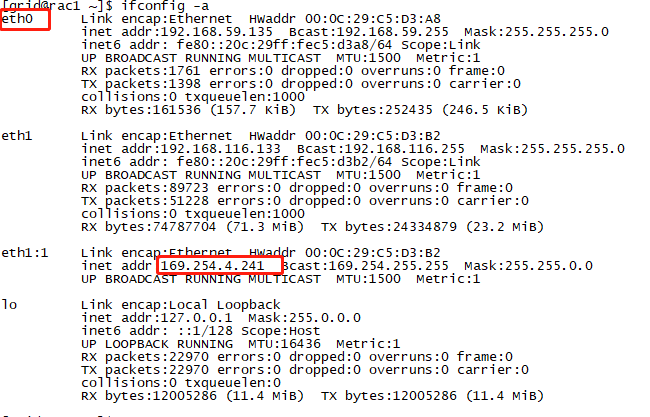
二节点的系统日志：



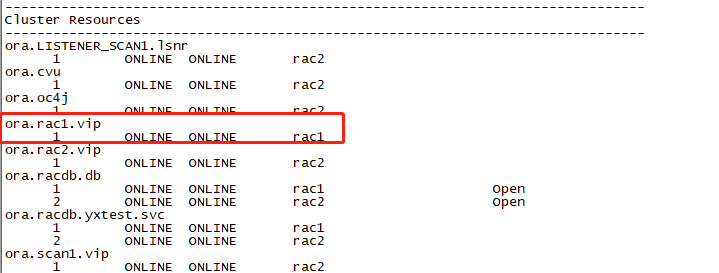
一节点的系统日志：



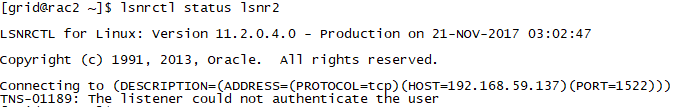
我们可以看到VIP是等集群都启动完了才漂回来的。这一点从一节点的ifconfig –a也能看的出来，HAIP都有了，VIP还没回来，如下是1节点crs启动过程中ifconfig的结果：



等系统日志显示VIP回来了以后我们再看集群：

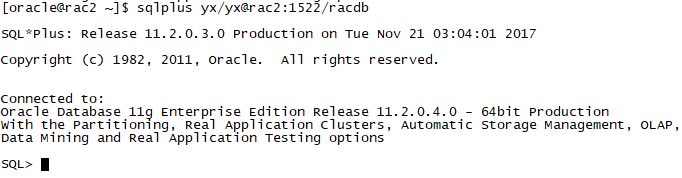


但是我们别忘了，我们再回去看看lsnr2的情况：



查看status会报错。

但是lsnr2还是能用的：



### Restricted Service Registration

### Software Components

一般来说，一个RAC数据库都存在2个或者2个以上的实例。这些实例拥有各自的memory和backgroud process。RAC数据库的数据库实例除了拥有所有非集群的实例的内存结构和后台进程，还会又属于RAC的额外的内存结构和后台进程。

RAC中的每个实例的SGA中都有自己的buffer cache。RAC通过cache fusion，逻辑的将这些buffer cache联合成一个让所有实例处理数据时仿佛是在一个单个的cache中处理数据。

注意，因为使用了cache fusion，所以RAC中每个实例的SGA的尺寸要求要大于非集群实例的SGA的尺寸要求。

为了确保在一个实例上的查询或事务可以更快的获得需要的block，RAC使用如下俩个进程：

1. Global Cache Service (GCS)
2. Global Enqueue Service (GES)

GCS和GES通过Global Resource Directory (GRD)来维护记录每一个数据文件和每一个被cache了的block的状态。

GRD的内容分布在所有RAC中活动的实例上，这会显著的增大SGA的size需求。

After one instance caches data, any other instance within the same cluster database can acquire a block image from another instance in the same database faster than by reading the block from disk. Therefore, Cache Fusion moves current blocks between instances rather than re-reading the blocks from disk. When a consistent block is needed or a changed block is required on another instance, Cache Fusion transfers the block image directly between the affected instances. Oracle RAC uses the private interconnect for interinstance communication and block transfers. The GES Monitor and the Instance Enqueue Process manage access to Cache Fusion resources and enqueue recovery processing.

在一个实例cache了一个block后，任何集群中的其他实例可以通过该实例获得block的image，这样比它们再次从磁盘中读取该block要更快。

就是说cache fusion在实例间移动current block要比从磁盘读block更快。

当需要读一个consistent block或者要修改一个block时，cache fusion将block image直接在相关的实例之间进行传输。

RAC是使用私网来进行block传输的。GES Monitor和Instance Enqueue Process管理对cache fusion资源和队列恢复处理的访问。

### Background Processes

ACMS: Atomic Controlfile to Memory Service (ACMS)

每个实例上都有ACMS进程，来保证数据库每个实例的SGA要么一起更新成功，要么就是当有失败发生时，一起放弃更新。

GTX0-j: Global Transaction Process

该进程对RAC环境中的XA global transactions提供透明的支持。数据库会自动根据XA global transactions的工作负载来调整该进程的数目。

LMON: Global Enqueue Service Monitor

LMON监控着集群中的global enqueues and resources。并且全局队列恢复的操作（global enqueue recovery）也是由LMON完成的。

LMD: Global Enqueue Service Daemon

LMD在每个实例中管理远程连入的资源请求。

LMS: Global Cache Service Process

LMS会将数据文件的状态和每一个被cache了的block记录进GRD（Global Resource Directory）。

LMS还控制着到其他实例的信息流。同时管理着不同实例间buffer cache中的block image的传输以及对global data block的访问。

LMS是Cache Fusion特性的一部分。

LCK0: Instance Enqueue Process

LCK0管理着non-Cache Fusion resource的请求，比如library and row cache的请求。

RMSn: Oracle RAC Management Processes (RMSn)

当有新的实例加入集群时，RMS进程会为其创建RAC相关的资源。

RSMN: Remote Slave Monitor

RSMN负责对remote instance上的background slave进程的创建和通信。这些后台从属进程是为了调节其他实例的上的进程的运行。

## 使用服务实现自动负载均衡

Automatic Workload Management with Dynamic Database Services。

服务可以将应用按特性、服务级别阈值和优先级进行分组。

应用功能用服务来划分成多块workload。比如说一个电子商务网站，可以分别为总账业务、订单业务、收账业务等等定义各自的服务。

一个服务可以在一个或者同时在多个实例上运行，每个实例上又可以定义多个服务。但是服务所服务的实例的个数对于应用来说是完全透明的。

连接数据库使用某服务的方式可以像这样：user\_name/password@SCAN/service\_name。

Oracle建议把连接串配置到tnsname里。

数据库建好后，oracle会自动为其创建一些服务。但是这些服务，oracle强烈建议不要使用这些服务去连接。在数据库建好后，我们可以手动创建各种服务，来区分限制各种业务对数据库的连接，从而达到更弹性的管理（greater workload management flexibility）。

Dynamic Database Services可以让我们更好的管理工作负载的分布。Dynamic Database Services可提供如下特性：

1. Services：  
   Oracle提供了一种很强大的进行自动负载均衡的工具，就叫做服务。这里所说的服务，是我们可以在RAC环境中定义创建的一个实体（entity），该实体可以用来进行业务分组、可以用来将工作均衡到当前最适合处理该工作的实例、并且可以应对各种计划中或计划外的状况和行为从而实现高可用。
2. High Availability Framework：  
   是RAC的一个组件，可以让数据库总是维持在运行状态。
3. Fast Application Notification (FAN)：  
   向RAC的应用或者客户端提供集群目前的状态信息，包括：集群状态变化和负载均衡建议事件，比如实例、服务、节点上的事件的起停。  
   FAN有俩种方式将events发布给客户端：Oracle Notification Service守护进程和Oracle Streams Advanced Queueing。但是从12.2以后开始，就只用Oracle Notification Service了。
4. Transaction Guard：  
   就是一个工具，可以为最多执行一次的事务（at-most-once execution of transactions）提供协议（Protocol）和接口（API），从而保证事务不会中途失败或者被重复提交。
5. Application Continuity：  
   就是当一个事务在执行过程中，遇节点故障（物理或者网络上的损坏），服务可以让该事务重新执行。对于应用而言，表现就是要做的事情被延迟完成，应用并不会知道后台到底发生了什么。
6. Connection Load Balancing：  
   自动把使用同一个服务连接数据库的连接均衡到运行该服务的实例上。
7. Load Balancing Advisory：  
   提供有关于当前数据库及其实例提供的服务级别的信息给应用。它基于定义服务时，给服务配置的管理策略（management policy）来给应用提供建议：现在应该让应用连到哪个实例会更好。Load Balancing Advisory events也是通过Oracle Notification Service发布给应用的。
8. Automatic Workload Repository (AWR)：  
   将服务级别的统计数据作为指标来进行跟踪。
9. Fast Connection Failover (FCF)：  
   这是ORACLE 客户端提供的一种：通过subscribe to FAN events（订阅FAN事件），来实现快速（rapid）故障转移的能力。
10. Runtime Connection Load Balancing  
    这是当应用发起连接数据库的请求时，Oracle Clients提供的一种可以智能的根据当前的数据库实例提供的服务级别从连接池中分配连接给应用。
11. Single Client Access Name (SCAN)  
    是RAC向ORACLE客户端提供的一个单一的名字，客户端可以配置连接串来连接这个名字（比如tnsnames中使用scan或者scan-vip）来进行对数据库的连接。同样可以实现自动负载均衡和连接失败转移。  
    这个名字贯穿集群一生，客户端使用该命字配置连接串，可以完全不需要知道RAC集群中节点的增删改的情况。

## Server Pools and Policy-Managed Databases

## Quality of Service Management

## Hang Manager

Hang Manager是ORACLE能够自动检测并解决system hangs的特性。该特性在RAC和单实例数据库中都有。

最初是在11.1引进的该技术，它识别system hangs，并且将相应的信息dump到trace文件中。从12.2开始，Hang Manager除了检测报告system hangs，同时还会自动的解决一些system hangs。

但是Hang Manager依然只能检测到dead lock，但并不能自动解决dead lock。

Hang Manager的功能如下：

1. 首先检测到一个hang，然后分析hang，确定hang的原因，最后决定一个解决hang的方法。
2. 自动产生trace文件，而不需要DBA去手动生成trace文件去交给My Oracle Supportd的大佬们去分析问题原因。变相缩短了数据库或者业务的downtime。
3. 周期性的巡检所有进程，并且自动的分析一小部分每次巡检都发现其上有持有资源的进程。Hang manager会自动忽略那些没有持有资源的进程。
4. 会考虑跨实例的hangs，比如hangs的holder是一个等待ASM实例响应的数据库实例
5. 检测reader nodes的进程，看其中是否有阻塞到hub nodes的进程。
6. 会考虑holder的QoS的设置。
7. 终止holder的进程来让后续的等待资源的进程继续工作，同时也阻止了一个hang。
8. 会在alert日志中记录ORA-32701的错误信息。

## Multitenant with Oracle RAC

主要就一点，每一个PDB可以在所有实例上open，也可以只在部分实例open。这些其实不是很重要，重要的我们可以通过dynamic database services来控制实例上PDB的访问，说白了就是使用哪些dynamic database services才可以访问哪些实例上open的哪些PDB。

## Database In-Memory and Oracle RAC

## Managing Oracle RAC Environments

### Administrative Tools

管理RAC可使用的管理工具如下：

1. Server Control Utility (SRVCTL)
2. Rapid Home Provisioning
3. Oracle Enterprise Manager
4. SQL\*Plus
5. Cluster Verification Utility (CVU)：  
   CVU是一个command-line工具，我们可以使用它来确认集群和RAC的许多组件和信息，比如：共享存储、网络配置、系统要求和ORACLE Clusterware，但这其中不包括操作系统的组和用户。  
   可以使用CVU对集群环境进行安装前检查和安装后检查。  
   CVU对于Oracle集群和RAC组件的安装前和安装中的检查极为有用。OUI会运行CVU在集群和RAC安装后对我们的环境进行确认。  
   在安装RAC集群前，最好安装并且使用CVU对环境进行检查，看看是否符合安装RAC的最小要求。另外，对于后续的一些管理工作，包括节点的增删，都应当使用CVU进行预检查。
6. DBCA
7. NETCA

### Monitoring

EMCC就不说了。

监控集群状态的动态性能视图比单实例又多了GV$（Global Views）开头的视图，几乎每个V$视图都有自己对应的GV$视图，区别就是GV$多了一列INST\_ID。也就是说V$查的就是本实例的，而GV$查的是所有实例V$的合集。

如果不是使用DBCA建的库，一定要记得执行catclustdb.sql脚本来创建GV$视图。

### Evaluating Performance

就是如果是单实例数据库转集群，如果应用在原本的环境中跑的就很顺利，那么在新的RAC环境中也不需要进行额外的变动。

对于Cache fusion，我们不需要对任何cache fusion相关的参数进行调整改变，同时应用界别的改变也是不需要的。

# Administering Storage in Oracle RAC

# 管理数据库实例和Cluster Databases